

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**МОДУЛЬНОЕ ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ:  
АНАЛИЗАТОР ПОКРЫТИЯ КОДА UNIT-ТЕСТАМИ ДЛЯ VISUAL  
STUDIO 2015**  
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы  
направления 02.03.03 Математическое обеспечение и администрирование  
информационных систем  
факультета компьютерных наук и информационных технологий  
Ефимова Ильи Сергеевича

Научный руководитель:

Доцент

\_\_\_\_\_

Е. В. Кудрина

подпись, дата

Зав. кафедрой:

Доцент, к.ф.-м.н.

\_\_\_\_\_

А. Г. Федорова

подпись, дата

Саратов 2016

## ВВЕДЕНИЕ

**Актуальность темы.** В каждой популярной модели разработки программного обеспечения (ПО) обязательно присутствует тестирование. Независимо от того, какая модель используется, эффективность тестирования является одним из важнейших факторов, определяющих стоимость и длительность разработки комплексов программ разной сложности.

Влияние тестирования на время разработки программного обеспечения и соответствующих затрат сложно переоценить: на фоне усиливающейся конкуренции software-производителей, ведущие мировые компании по разработке ПО организовали собственные отделы контроля качества (QA – quality assurance), которые обязаны следить, насколько исходный продукт соответствует техническому заданию, чтобы пользователь не испытывал неудобств в работе с приложением.

В 1979 была опубликована книга «Искусство тестирования программ». По мнению авторов книги, около половины времени разработки систем или программ тратится на их тестирование [1]. С момента издания данной книги появились новые системы разработки программного обеспечения, языки со встроенным инструментарием тестирования и программисты, владеющие быстрыми методиками разработки. Однако тестирование по-прежнему остается важной частью любого проекта по разработке программного обеспечения.

Принято выделять несколько уровней тестирования ПО – модульное тестирование, интеграционное тестирование, системное и приёмочное тестирование. Уровень тестирования определяет то, над чем производятся тесты: над отдельным модулем, группой модулей или системой, в целом [1]. Одним из основных уровней тестирования является модульное тестирование (unit-тестирование) – это процесс, позволяющий проверить на корректность отдельные модули исходного кода программы. Модуль (unit) представляет собой минимальную смысловую единицу исходного кода, пригодную для тестирования [2]. Это может быть процедура, функция или метод.

Существуют специальные инструментальные средства, которые позволяют оценить степень покрытия кода unit-тестами. От набора их функциональных возможностей, интерфейса и стоимости зависит, насколько удобно, полно и экономично можно реализовывать модульное тестирование. Следует отметить, что возрастает спрос не просто на специализированные средства unit-тестирования, а на расширения сред разработки, которые позволили бы осуществлять данный вид тестирования, не выходя из среды, что и определило цель бакалаврской работы – разработать анализатор покрытия кода unit-тестами для Visual Studio 2015 и предложить рекомендации по его применению на практике.

Поставленная цель определила следующие задачи:

1. рассмотреть различные модели разработки ПО и оценить роль тестирования в них;
2. познакомиться с историей развития теории тестирования программного обеспечения;
3. изучить модульное тестирование программного обеспечения;
4. проанализировать современные инструментальные средства для осуществления модульного тестирования;
5. предложить и реализовать собственный способ расширения возможностей среды Visual Studio 2015 для осуществления модульного тестирования

**Методологические основы** модульного тестирования представлены в работах Г. Майерса, Т. Баджетте, К. Сандлера [1], В.П. Котлярова, Т.В. Коликова, [6], А. Адеркина [10], И. В. Степаненко [28].

**Практическая значимость бакалаврской работы.** В ходе выполнения практической части бакалаврской работы было разработано собственное расширение для среды Visual Studio 2015 – плагин CoverageOfTest, позволяющий анализировать степень покрытия кода unit-тестами. Данный плагин распространяется бесплатно на русском языке и предназначен для свободного пользования всеми заинтересованными лицами.

**Структура и объём работы.** Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и 4 приложений. Общий объём работы – 61 страниц, из них 47 страниц – основное содержание, включая 35 рисунков и 1 таблицу, цифровой носитель в качестве приложения, список использованных источников информации – 28 наименований.

## **КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ**

**Первый раздел «Теоретические основы тестирования программного обеспечения»** посвящен описанию теоретических основ тестирования программного обеспечения в целом, а также более подробному изучению unit-тестирования. Раздел содержит несколько подразделов.

«Место и роль тестирования в различных моделях разработки программного обеспечения» включает в себя описание различных моделей разработки программного обеспечения, а также дана оценка роли тестирования в них. Рассмотрены каскадная, циклическая и V-образная модели.

«Развитие теории тестирования программного обеспечения» – подраздел посвящен изучению теоретических основ тестирования программного обеспечения и истории их развития.

«Модульное тестирование программного обеспечения» – в этом подразделе уделено особое внимание модульному тестированию.

Модульное тестирование (unit-тестирование) – это процесс, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея такого тестирования состоит в том, чтобы создавать unit-тесты – тесты для каждого нетривиальной модуля. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Рассмотрены платформы MSTest, NUnit, которые можно использовать для разработки unit-тестов для приложений, написанных для .NET Framework.

Подраздел «Сравнительный анализ современных инструментальных средств осуществления unit-тестирования» посвящён исследованию современных инструментальных средств для осуществления модульного тестирования – плагинов NCover, dotCover и инструментов покрытия кода PartCover и OpenCover. Проведённые исследования позволили сделать вывод о востребованности бесплатных анализаторов покрытия кода unit-тестами, расширяющих функционал Visual Studio.

**Второй раздел «Расширение возможностей Visual Studio для осуществления модульного тестирования»** посвящен реализации собственного плагина для среды разработки Visual Studio 2015 под названием CoverageOfTest, позволяющего анализировать степень покрытия кода unit-тестами. Раздел содержит три подраздела.

Следует отметить, что данная разработка, основывается на версии плагина CoverageOfTest, разработанного в 2015 году для Visual Studio 2013.

Перед новой версией плагина CoverageOfTest стоит задача обращения непосредственно к утилитах тестирования, чтобы оценить все ветви тестируемого кода. Дело в том, что некорректный модульный тест может, например, покрыть не все исходы условия if (если) в тестируемом методе, а это может привести к непредвиденным результатам.

Помимо этого, тестировщику будет удобней изучать так называемое «дерево тестов», где ему предоставляется информация о степени покрытия кода в виде цифр (отношение числа непокрытых строк к покрытым).

Плагин CoverageOfTest подсвечивает покрытые модульными тестами строки исходного кода. Пользователь данного расширения сможет визуально оценить степень покрытия по цвету строк исходного кода.

Пользователь плагина способен отключать возможность подсветки, или, например, при включенной подсветке смотреть на статистические

данные в информационной панели, которая сообщает пользователю, сколько строк покрыто и каково покрытие в процентном соотношении.

Для сборки информации о покрытии кода, плагин обращается к консольному приложению OpenCover, которое генерирует отчёты о покрытии кода, обращаясь к утилитам тестирования приложений на платформе .NET, вроде MSTest или NUnit.

В конце раздела предоставлена инструкция к эксплуатации плагина.

Разработанный автором данной работы плагин CoverageofTest легко встраивается в Microsoft Visual Studio 2015 и становится в ряд с остальными расширениями этой среды разработки. Благодаря чему разработчик программного обеспечения сможет самостоятельно сделать выводы о степени покрытия его кода unit-тестами и уделить больше внимания тем методам, которые ему предстоит покрыть. Чем больше покрытие кода, тем выше гарантия качества программного обеспечения. Если тестирующий ставит перед собой задачу обеспечить качество исходного продукта на наивысшем уровне, то ему следует стремиться к максимальному покрытию кода, то есть чтобы как можно больше строк считались покрытыми.

## **ЗАКЛЮЧЕНИЕ**

В ходе проделанной работы были решены все поставленные задачи, а именно: рассмотрены различные модели разработки программного обеспечения и дана оценка роли тестирования в них, изучены теоретические основы тестирования программного обеспечения, особое внимание при этом уделялось вопросам модульного тестирования, исследованы современные инструментальные средства для осуществления модульного тестирования – плагинов NCover, dotCover и инструментов покрытия кода PartCover и OpenCover. Проведённые исследования позволили сделать вывод о востребованности бесплатных анализаторов покрытия кода unit-тестами, расширяющих функционал Visual Studio.

Итогом бакалаврской работы стало разработанное расширение для среды Visual Studio 2015 – плагин CoverageOfTest, позволяющий анализировать степень покрытия кода unit-тестами. Данный плагин распространяется бесплатно на русском языке и предназначен для свободного пользования всеми заинтересованными лицами (ссылка для скачивания: <https://goo.gl/UkOAx8>).

По тематике бакалаврской работы были представлен доклад «Анализатор покрытия кода unit-тестами для Visual Studio» на VII Всероссийской (с международным участием) научно-практической конференции «Информационные технологии в образовании» «ИТО-Саратов-2015».

### **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Майерс, Г., Баджетт, Т., Сандлер, К. Искусство тестирования программ, 3-е издание The Art of Software Testing, 3rd Edition. — М.: Изд-во «Диалектика», 2012. — 272 с. — ISBN 978-5-8459-1796-6
2. Тестирование программного обеспечения: модульные тесты [Электронный ресурс]. URL: [openquality.ru/software-testing/unit-tests.php](http://openquality.ru/software-testing/unit-tests.php) (дата обращения: 11.05.2014)
3. Брукс, Ф.П. Как проектируются и создаются программные комплексы. — М.: Мир, 1979.
4. Модели жизненного цикла программного обеспечения // Habrahabr [Электронный ресурс]. URL: [habrahabr.ru/post/111674/](http://habrahabr.ru/post/111674/) (дата обращения: 21.04.2015).
5. Макконнелл, С. Rapid Development: Taming Wild Software Schedules. Изд-во «Microsoft Press», 1996. – ISBN 1-55615-900-5.
6. Котляров, В.П., Коликова, Т.В. – Основы тестирования программного обеспечения. – Изд-во «Бином. Лаборатория знаний», 2009. – 285 с. – ISBN 5-94774-406-4

7. Fundamentals of the V-Modell [Электронный ресурс] // IABG [Электронный ресурс] : [сайт]. URL: [http://www.v-modell.iabg.de/kurzb/vm/k\\_vm\\_e.doc](http://www.v-modell.iabg.de/kurzb/vm/k_vm_e.doc) (дата обращения 06.03.2016). Загл. с экрана. Яз. рус.
8. Форсберг, К. Муз, Х., The Relationship of Systems Engineering to the Project Cycle [Электронный ресурс] // Первый ежегодный симпозиум национального совета по системной инженерии (октябрь 1991 года) [Электронный ресурс] : [сайт] URL: [http://www.csm.com/repository/model/rep/o/pdf/Relationship of SE to Proj Cycle.pdf](http://www.csm.com/repository/model/rep/o/pdf/Relationship%20of%20SE%20to%20Proj%20Cycle.pdf) (дата обращения 27.02.2009). Загл. с экрана. Яз. рус.
9. Fundamentals of the V-Modell [Электронный ресурс] // IABG [Электронный ресурс] : [сайт]. URL: <http://v-modell.iabg.de/v-modell-xt-html-english/f3ffba5de1675.html> (дата обращения 06.03.2016). Загл. с экрана. Яз. рус.
10. Адеркин, А. Тестирование программного обеспечения [Электронный ресурс]. [Электронный ресурс] : [сайт]. URL: <http://aderkin.ru/> (дата обращения: 24.04.2013)
11. Определение и принципы тестирования [Электронный ресурс]. URL: <http://www.rae.ru/monographs/141-4632> (дата обращения: 19.04.2013).
12. Google Trends [Электронный ресурс]. URL: <http://www.google.com/trends?q=nunit%2C+mbunit%2C+mstest%2C+xunit> (дата обращения: 09.03.2015)
13. Comparing the MSTest and Nunit Frameworks [Электронный ресурс]. URL: <http://blogs.msdn.com/b/nnaderi/archive/2007/02/01/mstest-vs-nunit-frameworks.aspx> (дата обращения: 09.03.2015)
14. Киан, Д. New for Visual Studio 2008 – Code Metrics // MSDN [Электронный ресурс]. URL: [blogs.msdn.com/b/codeanalysis/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx](http://blogs.msdn.com/b/codeanalysis/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx) (дата обращения: 15.05.2014).
15. Developer Tools Microsoft Visual Studio 2013 SDK // Microsoft Download Center [Электронный ресурс]. URL: <https://www.microsoft.com/en-us/download/details.aspx?id=40758> (дата обращения: 26.05.2015)

16. Unisys Corporation, EDT Text Editor Reference Manual. – Cinnaminson, New Jersey: 1975. – 363 с
17. NCover [Электронный ресурс]. URL: <https://www.ncover.com/> (дата обращения: 29.05.2016)
18. Free for students: Professional developer tools from JetBrains // JetBrains [Электронный ресурс]. URL: <https://www.jetbrains.com/student/> (дата обращения: 28.05.2016)
19. partcover.net4 Download Packages // GitHub [Электронный ресурс]. URL: <https://github.com/sawilde/partcover.net4/downloads> (дата обращения: 27.05.2016)
20. Introduction to PartCover - a short manual // Kowalczyk [Электронный ресурс]. URL: <https://blog.kowalczyk.info/article/64oh/Introduction-to-PartCover-a-short-manual.html> (дата обращения: 27.05.2016)
21. Yefimov, Mono in VS2015 // OpenCover.Issues [Электронный ресурс]. URL: <https://github.com/OpenCover/opencover/issues/583> (дата обращения: 28.05.2016)
22. Руководство по разработке модулей расширений на C# для Visual Studio 2005-2012 и Atmel Studio // Хабрахабр [Электронный ресурс]. URL: <http://habrahabr.ru/company/pvs-studio/blog/192486/> (дата обращения: 21.05.2015)
23. VSPackages // MSDN [Электронный ресурс]. URL: <https://msdn.microsoft.com/ru-ru/library/bb166424.aspx> (дата обращения: 21.05.2015)
24. Add-ins and Extensibility // MSDN [Электронный ресурс]. URL: <https://msdn.microsoft.com/en-us/library/bb384200%28v=vs.110%29.aspx> (дата обращения: 22.05.2015)
25. Автоматизация и расширение среды для Visual Studio // MSDN [Электронный ресурс]. URL: <https://msdn.microsoft.com/ru-ru/library/xc52ske4.aspx> (дата обращения: 22.05.2015)

26. Developer Tools Microsoft Visual Studio 2013 SDK // Microsoft Download Center [Электронный ресурс]. URL: <https://www.microsoft.com/en-us/download/details.aspx?id=40758> (дата обращения: 26.05.2015)
27. Extensibility Projects // MSDN [Электронный ресурс]. URL: <https://msdn.microsoft.com/ru-ru/library/vstudio/6t415b78%28v=vs.110%29.aspx> (дата обращения: 12.04.2015)
28. Степанченко И. В. Методы тестирования программного обеспечения: Учеб. пособие / ВолгГТУ, Волгоград, 2006. – 74 с. – ISBN 5-230-04563-9.